

COMAL-80 Standardisation & Development Group Meetings
University of Copenhagen, 2nd & 3rd December 1983.

NOTE

As the meetings of the two groups were interleaved throughout the two days, these minutes cover both group meetings.

1.0 Attendance

1.1 Members of the Standards Group:

Borge Christensen (Chairman), Kevin Ryan (Secretary), Roy Atherton (Chairman of the Comal80 Development Group), Jens Erik Jensen, Arne Christensen, Erik Jeppeson. New members admitted (see 3.2 below) David Christensen, Anders Anstad, Bengt Nillson.

1.2 Observers :

Bernodette O'Leary, Rody Ryan, Neils Roch, Mogens Pella, Lars Lauensten.

2.0 Introduction

Borge Christensen (BC) welcomed the participants and said they were meeting at a critical time for Comal. Many structured Basics were on the market and to succeed we must make progress quickly. He felt the adoption of Comal80 by Commodore for their 64 machine would be a help.

Arne Christensen said that as there had been no preliminary circulation of proposals there was nothing ready to be added to the Kernel. Borge Christensen felt that decisions were essential, and informed the meeting that whatever its outcome he (Borge Christensen) would no longer be Chairman. Roy Atherton suggested that some decisions should be made, even if they were not optimal. Rody Ryan said we should not be discouraged. He believed Comal had a chance to benefit from the conflict between Pascal and Basic. All of us could benefit by co-operating to bring Comal to the market.

COMAL-80 Standardisation & Development Group Meetings
University of Copenhagen, 2nd & 3rd December 1983.

3.0 Administration & Agenda

3.1 Costs

It was suggested that participants should contribute to the administrative costs of the Group. Borge Christensen said Comal had always been a "low-budget" language. Bengt Nilsson felt companies should contribute.

3.2 New members

There were three new implementers represented : Acornsoft, Esselte-Studium, and Dansk Data Elektronik. When it was proposed that all three be admitted as full members of the Comal80 Standardisation Group, Mogens Pelle said that of the representatives might not be able to commit their companies to decisions made at the meeting. It was agreed to admit all three, represented respectively by : Dovid Christensen, Bengt Nilsson and Anders Ansted.

3.3 Agenda

Borge Christensen suggested, and it was agreed, that a Comal80 Development Group meeting should first be held. It was also agreed to postpone consideration of the date and location of the next meeting.

The Comal80 Standardisation Group adjourned at 2pm.

COMAL-80 Standardisation & Development Group Meetings
University of Copenhagen, 2nd & 3rd December 1983.

COMAL-80 Development Group Meeting
University of Copenhagen, Friday 2 December 1983.

4.0 Introduction

Borge Christensen(BC) opened the meeting and introduced the three new members. Bernadette O'Leary(BOL) agreed to act as secretary due to the departure of Brendan Lynch from the group. Roy Atherton (RA) took the chair and it was agreed that the meeting would attempt to finish the definition of the kernel. There were no objections to the minutes of the previous meeting and he opened discussion on matters arising from these minutes.

5.0 Matters Arising

5.1 Scope and Binding.

Jens Erik Jensen(JEJ) suggested we take simpler matters first. Kevin Ryan(KR) pointed out that it was not discussed previously because it was too complex and since it was two years since it was first mentioned, it was time we knew the position. RA made the point that David Christensen(DC) had not read the paper and therefore could not contribute, so it should be postponed. KR brought to notice the discussion of binding in the Acornsoft paper. Finally it was decided to discuss EXEC - a simple item, before the break.

5.2 EXEC

RA presented the choices (1)Obligatory, (2)Optional, or (3)Definitely Absent.

Arne Christensen(AR) was not in favour of (2).

DC argued that if LET in the assignment statement is optional, EXEC should also be optional since they were the same type of statement.

AC said that the same argument could hold for the DATA statement.

KR said that without an EXEC any rubbish can be interpreted as a procedure call.

COMAL-80 Standardisation & Development Group Meetings
University of Copenhagen, 2nd & 3rd December 1983.

BC only liked choices (2) and (3) in particular (2). He argued that optionality was necessary in order to use procedures in command mode. For example, in Graphics one should be able to say TURN as they do in LOGO, not EXEC TURN which is cumbersome. This presents the user with a strong abstract tool.

Mogens Pelle(MP) argued that procedures should not be used in this manner. For this type of application you need a different structure in the language, one should not use procedures to use new hardware facilities.

BC claimed that EXFC was a nuisance at times and it was only introduced initially during the transition from BASIC to COMAL.

MP said that there should be another way of implementing new keywords, with EXEC the user can always find the body of a procedure in the code.

RA remarked that in the new structured BASICs, procedures have no parentheses and no EXFC and they seem to work very well.

JEJ said it was important to keep parentheses and EXEC so the user could distinguish between extended keywords from machine language routines and procedure calls which can be listed.

AC thought that JEJ's argument was a good one in favour of leaving out parentheses, because the way procedures are implemented should not matter, they are all called the same way. He was in favour of choice (3).

BC gave an example of where it was necessary to have parentheses:

```
a := zap 1,pow 2,3
a := zap (1,pow(2),3)
a := zap (1,pow(2,3))
```

He made the point that COMAL should be read clearly, one should not have to sort out ambiguities.

RA suggested that a vote should be taken on: no EXFC, no parentheses in procedures and optional parentheses in functions.

KR argued that we were in no position to make a decision on this. We should only vote on what AC proposed since he was the only one who wrote a paper giving his

COMAL-80 Standardisation & Development Group Meetings
University of Copenhagen, 2nd & 3rd December 1983.

opinions, otherwise we could make a bad mistake. He suggested that before making a change each member should write a paper outlining their stand. He pointed out that AC had only proposed Option D (No EXFC and no Parentheses) from his paper, and no proposals had been received from other members.

RA proposed that we should start with a vote on (D) and that we make AC's paper the centre of discussion.

Erik Jeppesen(EJ) said that others had not sent out proposals as they are quite satisfied with the present situation.

A vote for (D) was defeated 4-2. AC proposed (A), (EXEC and parentheses both obligatory). (A) was also defeated 5-1.

DECIDED to leave the Kernel unmodified. EXFC optional and parentheses obligatory.

Meeting was then adjourned for 30 mins.

5.3 Scope and Binding.

RA brought to notice the four papers by the following authors:

- (1) Arne Christensen.
- (2) A/S Regnecentralen. (distributed at meeting)
- (3) T.C.D. (notes with Agenda)
- (4) Acornsoft paper (Page 21)

Presently no firm rules exist, he suggested that each author should state their position.

AC said that this was a complicated matter and we should be very careful when discussing new proposals in case we overlook something.

Rody Ryon(RR) suggested that we have a preliminary meeting and not make a decision right now.

JFJ agreed and said we should think over this subject and decide the following day.

RA suggested that each member make a statement.

AC started by outlining the problems of scope and binding so that everyone could understand what they were. He illustrated that the complications arose because COMAL

COMAL-80 Standardisation & Development Group Meetings
University of Copenhagen, 2nd & 3rd December 1983.

programs can have both OPEN and CLOSED procedures. Conflicts arise between OPEN/CLOSED procedures, Static/Dynamic binding, Nesting/No Nesting of procedure definitions.

KR said that the problems arose with binding in OPEN and CLOSED procedures.

RC put forward the following example and asked for comment:

```
t$ := "AND"
printout("NONSENSE")
|
|
PROC printout(t$)
|
|   output
|   |
ENDPROC
|
PROC output
|   PRINT t$
ENDPROC
```

He asked what value of t\$ would be printed ?

EJ argued that one should know from the main program which arguments will be altered by a procedure.

KR argued that it is easier to understand static programs, and it is harder to apply logic to dynamic programs. Metanik have the same stand on static binding as TCD.

DC argued that the most important thing is how a program is executed. He claimed that people work through a program with static binding, but with dynamic one can see what is going on.

KR said that printing the most recent value of a variable is very dangerous. If a programmer uses parameters there is no problem and no dispute between the different implementations.

RC said with regard to his earlier example, that he explains "t\$" as being "alive", therefore he explains dynamic binding to his students.

RA recognized the existence of two camps: four implementors being in favour of dynamic binding and two in favour of static binding.

AC offered to explain a "trick" he used in his first implementation which could bring the two camps together. He suggested that we make a few rules: The parameters of an OPEN procedure which calls another OPEN procedure are not available to the called OPEN procedure. There is still a problem if calls to OPEN and CLOSED procedures are mixed. The following restriction resolves this problem:

An OPEN procedure can call a CLOSED procedure.

A CLOSED procedure cannot call an OPEN procedure.

KR noted that the advantage of this restriction is that programs will then be able to run on all machines. He then formalised AC's rules

(1) Procedures cannot have access to the calling procedures parameters.

(2) OPEN procs can call a CLOSED.

CLOSED cannot call an OPEN.

AC said that if programs adhere to these rules, dynamic binding is the same as static and binding rules are immaterial.

RA wanted to find a solution that would not burden less able programmers with rules.

KR said that the reason TCD choose static binding was because it was simpler for beginners, not on the basis of implementation.

BC said that from an educators point of view, he would much prefer static.

KR suggested that we do not mention the question of binding to users, just introduce the two rules.

F.I said that one of the reasons they choose dynamic binding was that it was not enough to have structured programs, but also one needs modular programs. They want nesting as an extension, so they do not want anything in the kernel that might disallow it. For examples:

PROC q CLOSED

PROC p

|


```
ENDPROC p
p
|
ENDPROC q
```

They want to be able to call procedure "p".

AC suggested that we extend the second rule, to be:
(2) A CLOSED cannot call on OPEN that is not defined inside the calling CLOSED procedure.

DC did not like the idea of parameters being treated differently to variables, he was not happy with rule (1).

KR said that rule (2) was good programming advice, but was not happy with rule (1).

RA said that if we accept rule (1) with the right kind of phrasing we make the kernel sensible and leave open the possibilities of extensions. If we cannot accept rule (1), we must make the choice between static and dynamic binding.

DC suggested that not accepting rule (1) could be an extension.

KR remarked that relaxing a rule in the standard is an implementation decision and it is desirable to advise people to follow the standard. Rules are worth having even if they are bad rules, in order to enforce a standard.

RA proposed that the four implementers involved: KR, AC, DC and EJ should draft the rules overnight, to be agreed at the Standards meeting the following day.

5.4 Rounding vs Truncation.

EJ said that after last meeting they now use rounding. DC truncates since all BASIC's truncate. AC remarked that we should make COMAL better than any BASIC. DC was happy to concede to rounding.

AGREED that all implementers round rather than truncate and to recommend that this be in the Kernel.

5.5 Definition of DIV and MOD.

AC pointed out the differences of the three implementations. He referred to the cases (a), (b) and (c) from his paper. The safe case being:

$$\begin{array}{l} X \text{ (DIV) } Y \quad X, Y > 0 \\ \text{ (MOD)} \end{array}$$
$$X := N * Y$$

KR would accept DIVo, but not MODo.

AC liked (a) or (c).

EJ claimed he had complaints from users regarding the (c) version, so he changed to (a).

RC said we should make a note that they should be used with positive numbers only.

RA remarked that three companies use (a), EJ could not remember which they used.

AC said it would be difficult to change as old programs would go berserk.

RA proposed that this discussion be discontinued till the following day. Decide on this issue tomorrow, together with DIV and MOD.

5.6 Short Forms.

Decided to leave discussion till the following day.

5.7 Declaration of String Variables.

RC proposed that a default declaration of a simple string variable be allowed. If a string variable is used without first being DIMed, its length is automatically set to 40. He suggested that people should think about this and discuss it the following day and may be put it in as an extension or in the kernel.

DC suggested we look at page 17 of the Acornsoft paper where a problem with formal string parameters is detailed.

COMAL-80 Standardisation & Development Group Meetings
University of Copenhagen, 2nd & 3rd December 1983.

RA brought the meeting to a close. The following day the questions of: Binding, OLV and MOD, string declarations and short forms were to be presented to the Standards Group.

Comal80 Development Group Meeting
(resumed 9.45am Sat. 3rd December)

5.8 Binding

After a discussion on the implications of making static OR dynamic obligatory it was

AGREED that the following three points should be recommended for inclusion in the Kernel:

1. A Note of Advice be incorporated stating that "to ensure portability of software programs should NOT rely on their formal parameters being available to called programs."
2. Procedure names must be IMPORTed into CLOSED procedures.
3. OPEN procedures can NOT be IMPORTed.

The following example program illustrates the binding problem. The use of "y" in PROC B is ambiguous.

```
y:=3
a(1,2)
PROC A(x,y)
    B
ENDPROC
PROC B
    Print y
ENDPROC
```

5.9 Test Programs

To improve semantic compatibility Kevin Ryan(KR) asked that each implementor send him some test programs which were believed to be in accordance with the Kernel. These would be circulated together with the results obtained and

COMAL-80 Standardisation & Development Group Meetings
University of Copenhagen, 2nd & 3rd December 1983.

discrepancies should be reported to the secretary, who would put these on the agenda for the next meeting.

5.10 Re-use of Names

KR raised the question of using the same names for (e.g.) a label and a procedure. TCD had a proposal on this which would be considered at the Comal80 Standardisation Group meeting.

Proposal : "The first use of a name defines its type".

5.11 DIV and MOD

It was reported that Metonic and RC both had "DIVo" and "MODo", while Unicomal had "DIVb" and "MODb".

5.12 Strings - Default Length

Borge Christensen (BC) argued that if dynamic strings were NOT implemented a Comal should provide a default length when a string variable was used without ever being DIMed. There was a discussion on the relative merits of static and dynamic string handling. BC said we should state that dynamic was preferred (and not for technical reasons). DC said he had worked from the Kernel syntax only and so was unaware of this. Roddy Ryan (RR) emphasised that this showed the need for wide publicity of the Groups activities. It was agreed that a press release would be produced by the Secretary. After further discussion it was

AGREED that :

1.

A note saying that dynamic string handling was preferable would be incorporated in the Kernel.

2.

An extension to the Kernel would state that if dynamic strings were NOT implemented a default length of 40 would be given to an unDIMed string variable.

COMAL-80 Standardisation & Development Group Meetings
University of Copenhagen, 2nd & 3rd December 1983.

The Comol80 Development Group meeting adjourned at 11.15am.

Comal80 Standardisation Group Meeting
(resumed 11.48am Sat 3rd Dec.)

6.0 Agenda

The meeting reconvened to deal with the outstanding proposals from the Comal80 Development Group .

6.1 EXEC

AGREED no change in the standard.

6.2 Scope and Binding

With minor amendments the TCD proposal on naming was accepted as was (subject to ratification) the binding suggestions from the Comal80 Development Group .

AGREED :

SCOPE: Within a given scope, the first use of a name, whether as a procedure name, function name, label or variable name (whether DIMed or not) defines the names type. Any subsequent attempt to redefine this name is an error.

BINDING

1. A Note of Advice to be incorporated stating that "to ensure portability of software, users programs should NOT rely on their formal parameters being available to called programs."
2. Procedure names must be IMPORTed into CLOSED procedures.
3. OPEN procedures can NOT be IMPORTed.

6.3 DIV and MOD

Implementors stated their preferences from the list of options circulated by Arne Christensen after the March meeting. On the proposal of Mogens Pelle the option MODc and DIVc (with MOD defined only for positive y) was adopted by 7 votes to 1.

AGREED

The DIV function is defined to take two arguments x and y and to return the next lowest integer less than or equal to the result of dividing x by y.

The MOD function also takes two arguments x and y (y must be positive) and returns $(x - (x \text{ DIV } y) * y)$. These definitions of MOD and DIV are such as to give the following results :

36 MOD 5 is 1	36 MOD -5 is undefined
-36 MOD 5 is 4	-36 MOD -5 is undefined
35 MOD 5 is 0	35 MOD -5 is undefined
-35 MOD 5 is 0	-35 MOD -5 is undefined
36 DIV 5 is 7	36 DIV -5 is -8
-36 DIV 5 is -8	-36 DIV -5 is 7
35 DIV 5 is 7	35 DIV -5 is -7
-35 DIV 5 is -7	-35 DIV -5 is 7

6.4 Formal String Parameters.

DC raised the question of how the length of a formal string parameter is to be determined, and advocated the method shown on p.18 of the Acornsoft paper. This would involve an OF clause within the parameter list. AC, EJ and Lars Lauersen(LL) all disagreed with this. After some discussion it was

AGREED to leave the Standard unchanged.

6.5 Dynamic & Static Strings.

The proposals from the Development Group were generally acceptable. An additional rule concerning the use of "OF length" with dynamic strings was accepted (subject to ratification).

AGREED to add to the Kernel :

Dynamic string handling is preferable and if it is implemented, the "OF length" clause in a DIMension statement sets a maximum length for a string.

6.6 Warning Note.

Before adjourning for lunch MP suggested that the Draft Standard should include a "Warning Note" for any prospective implementors mentioning areas which are currently under discussion by the group. This was agreed.

7.0 Administration

The Standards meeting resumed at 3.40pm to deal with two items.

7.1 Chairmen of Standardisation Group.

As Borge Christensen had resigned (but would remain as a member of the Development Group) nominations were sought for Chairman. Roy Atherton agreed to act as Chairman of both groups until the next meeting, but pointed out that his workload would not allow this to continue. RA suggested that a well-known figure (e.g. D Borron) from the computing world be sought to take the chair. It was agreed that the names of any available persons would be submitted to the secretary before the next meeting.

7.2 Date & Location of Next Meeting.

It was agreed after discussion that the next meeting would be held in or near London (possibly Reading) on the 13th and 14th of April 1984.

Resumption of Development Group meeting

COMAL-80 Standardisation & Development Group Meetings
University of Copenhagen, 2nd & 3rd December 1983.

8.0 Agenda.

The Development meeting resumed to deal with Short Forms and to try to reach an agreement that could be ratified by the Standardisation Group.

8.1 Short Forms of IF, FOR, WHILE & REPEAT.

RA opened the discussion by referring to the minutes of the previous meeting. MP felt that the decision on whether to include these forms in the Kernel should only be made after considering the criteria for inclusion. As they do not add features to the language he felt they should NOT be included. BC said the short FOR was useful in teaching iteration.

DC did not like them because they hide program structure, but agreed they would be easy to teach.

RA had prepared a paper favouring their inclusion and this paper was copied and distributed.

KK asked RA if he favoured the FOR and IF more than the other two.

MP said shorts were only a special case, but DC said they were an important special case.

The chairman asked for a statement of the current situation. After this it was agreed to take the IF and the short loops as two separate issues. The inclusion of the IF in the kernel was recommended by 9 votes to 2. The inclusion of All three loops in the kernel was recommended by 10 votes to 3.

Resumption of Standardisation Group meeting

9.0 Agenda

This meeting was resumed at 5pm to consider the recommendations on short forms.

9.1 Short Forms of IF, FOR, WHILE & REPEAT.

With six implementors voting (plus the Chairman) under the rules of the Group, 4 out of 7 votes were required to carry a proposal. The proposal that the short IF be removed from the Kernel was defeated by 4 votes to 1. The short loops were taken individually but only the short FOR had any support for inclusion in the Kernel. This was agreed by 4 votes to 1. The other short forms were unanimously accepted as standard extensions.

AGREED

1. The short IF remains in the Kernel.
2. The short FOR is added to the Kernel.
3. The short WHILE and short REPEAT are standard extensions.

Resumption of Development Group meeting

10.0 Agenda

The meeting resumed at 5.30pm and as there was very little time a number of small points were dealt with rapidly.

10.1 RETURN

A semantic note is required to distinguish between the behaviour of Procedures and Functions. Secretary to draft this.

10.2 GOTO out of a Structured Statement.

It was confirmed that this is allowed.

COMAL-80 Standardisation & Development Group Meetings
University of Copenhagen, 2nd & 3rd December 1983.

10.3 STOP with a Message.

It was agreed that this would be an extension for now.

11.0 Administration

11.1 Costs.

It was agreed that each manufacturer would pay 200 pounds sterling towards the costs of the Group. Kevin Ryan would collect and account for this money.

11.2 Manuals.

In addition to the test programs it was agreed that manufacturers should provide copies of their manuals to one another to check for incompatibilities.

The meeting closed at 6.00pm